

REMARKS

Claims 1-9, 11-19, and 21-29 were pending at the time the Office Action was issued, with claims 10, 20, and 30 having previously been withdrawn from consideration.

Claims 1-9, 11-19, and 21-29 were rejected under 35 U.S.C. § 103(a) as being unpatentable in an Office Action mailed on April 20, 2007.

Independent claims 1, 11, and 21 are currently amended.

No claims are presently canceled.

Thus, claims 1-9, 11-19, and 21-29 remain pending.

Summary of Interview

The Examiner, Mr. Pierre Michel Bataille, and was kind enough to conduct a telephone interview with applicants' representative on Wednesday, July 11, 2007, to discuss the claims.

Applicants' representative submitted a draft of proposed amended claims and supporting remarks, which the Examiner considerately took the time to review before the interview.

Proposed amendments to independent claims 1, 11, and 21, which stand rejected under 35 U.S.C. § 103(a), were discussed relative to the cited references including U.S. Patent Application Publication No. 2004/0221120 A1 of Abrashkevich et al. in view of U.S. Patent No. 7,158, 991, to Kekre et al.

The Examiner asked how the proposed amended claims were supported in the specification. Applicants' representative identified existing support in the specification for a timestamp stored in a memory block, and also referenced a few originally-filed dependent claims

that disclosed the nature of the subject matter. Applicants' representative indicated he would seek to amend the specification to include what was recited in the originally-filed claims.

The Examiner indicated that, if he is satisfied that the amendments are supported by the original disclosure, that the claims as amended distinguish over the references cited, and the current rejection under 35 U.S.C. § 103(a) based on the currently cited references would be withdrawn.

Applicants and their representative thank the Examiner for his time and consideration.

Amendment to the Specification

The foregoing amendment is made to incorporate content into the specification that was recited in originally-presented claims 10, 20, and 30. MPEP § 608.01(I) makes clear that content included in the originally-presented claims constitutes part of the disclosure of a patent application:

In establishing a disclosure, applicant may rely not only on the description and drawing as filed but also on the original claims if their content justifies it.

Where subject matter not shown in the drawing or described in the description is claimed in the application as filed, and such original claim itself constitutes a clear disclosure of this subject matter, then the claim should be treated on its merits, and requirement made to amend the drawing and description to show this subject matter. The claim should not be attacked either by objection or rejection because this subject matter is lacking in the drawing and description.

Thus, in the interest of completeness of the specification, as allowed by the MPEP, applicants have incorporated content from originally-filed claims 10, 20, and 30 into the specification. Applicants attest that inclusion of this content introduces no new matter into the application.

Replacement Drawing Sheet

Applicants respectfully request that the replacement drawing sheet included in the Appendix of this response be entered to replace current Fig. 4. The previously described amendment to the specification references a “timestamp” that was included in original Fig. 4, but for which no reference number was presented in Fig. 4. The replacement drawing sheet adds reference number “470” to Fig. 4. No other changes are made to Fig. 4. Again, applicants attest that inclusion of this content introduces no new matter into the application.

Rejections under 35 U.S.C. § 103(a)

Claims 1-9, 11-19, and 21-29 were rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent Application Publication No. 2004/0221120 A1 of Abrashkevich et al. (hereinafter “Abrashkevich”) in view of U.S. Patent No. 7,158, 991, to Kekre et al. (hereinafter “Kekre”). Applicants respectfully traverse the rejections. Independent claims 1, 11, and 21 are currently amended to further clarify the distinctions between the claims and the references cited.

In the interest of reducing the number of issues for the Examiner to consider in this response, the following discussion focuses on independent claims 1, 11, and 21. The patentability of each remaining dependent claim is not separately addressed. However, applicants' decision not to discuss the differences between the cited art and each dependent claim should not be considered as an admission that applicants concur with the Examiner's conclusion that these dependent claims are not patentable over the disclosure in the cited references. Similarly, applicants' decision not to discuss differences between the prior art and every claim element, or every comment made by the Examiner, should not be considered as an admission that applicants concur with the Examiner's interpretation and assertions regarding those claims. Indeed, applicants believe that all of the dependent claims patentably distinguish over the references cited. Moreover, a specific traverse of the rejection of each dependent claim is not required, because dependent claims are patentable for at least the same reasons as the independent claims from which the dependent claims ultimately depend.

Claim 1 as amended is reproduced on the next page for the convenience of the Examiner:

1. (Currently Amended) A method for tagging an allocable memory block, comprising:

determining the identity of a routine performing one of requesting the allocable memory block, requesting the size of the allocable memory block, and freeing the allocable memory block;

generating an identifier for the routine;

storing the identifier in the allocable memory block; and

storing a timestamp within the allocable memory block, wherein the timestamp is configured to indicate indicates a time when one of the requesting and the freeing of the allocable memory block is performed;

the requesting of the allocable memory block is performed unless the timestamp indicates a time when the allocable memory block is freed; and

the freeing of the allocable memory block is performed unless the timestamp indicates a time when the allocable memory block is requested,

such that, upon detection of a memory usage error involving the allocable memory block, the identifier for the routine and the timestamp provide information usable in determining whether the routine is causing memory errors.

A comparison of claim 1 with the cited references makes clear that claim 1 is patentable over the prior art. The Office Action concedes that Abrashkevich fails to teach storing a timestamp within the allocable memory block, but asserts that Kekre overcomes this admitted shortcoming of Abrashkevich. While applicants acknowledge that Kekre has a temporal aspect that Abrashkevich lacks, for at least four reasons, Kekre does not and cannot make up for the shortcomings of Abrashkevich.

First, the Office Action fails to present a *prima facie* case of obviousness because a reasonable person of ordinary skill in the art would not have looked to Kekre to make up for the admitted shortcomings of Abrashkevich. Claim 1 expressly recites “storing a timestamp within the allocable memory block . . . such that, upon detection of a memory usage error involving the allocable memory block, the identifier for the routine and the timestamp provide information usable in determining whether the routine is causing memory errors.” Like Abrashkevich, the

objective of claim 1 is the diagnosis of memory errors. Kekre, on the other hand, has nothing to do with diagnosis of errors of any kind

Kekre is directed to objectives and operational layers that are different than the functions or layers to which either Abrashkevich or claim 1 are directed. Kekre describes a system for tracking the time at which changes were made to portions of a body of data resulting in a “temporal volume . . . that maintains non-present data in addition to the present data.” Kekre, Column 1, Lines 50-51. In other words, Kekre maintains changes applied to a body of data over time to, in effect, maintain versions of the data existing at different points of time:

A temporal volume may maintain the history of data stored on it, thus providing a way for the application to retrieve a copy of the data at any time in the past. In a temporal volume, *whenever a block of data is to be changed, the existing block is first preserved, and then the new data is overwritten. The old versions of a block are maintained even when the block is deleted by the application from the data.* This achieves the effect of maintaining copies of one or more states of the data in the past.

Kekre, Column 1, Lines 51-60 (emphasis added). Thus, Kekre tracks when data is changed, to be able to retrieve past versions of the data existing at different points in time.

By contrast, Kekre is *not* concerned with identifying when memory blocks were allocated or freed to determine if a routine is causing memory errors. In fact, Kekre is not concerned with memory blocks at all; Kekre is concerned with an application-level function, whereas Abrashkevich and claim 1 are directed to a lower-level operation concern. By analogy, Kekre is directed to allowing a user to rewind a videotape, whereas Abrashkevich and claim 1 are directed to the mechanism by which images are stored on the tape. In short, Kekre and Abrashkevich are directed to different objectives, and a reasonable person would not have reached to Kekre or thought to try to reach for Kekre to overcome the shortcomings of Abrashkevich. Respectfully,

only with the benefit of hindsight would one of ordinary skill in the art, or even extraordinary skill in the art, have combined Abrashkevich's a memory block allocation diagnostic aid with Kekre's data versioning tool. The Office Action thus fails to present a *prima facie* case of obviousness, and the rejection should be withdrawn against the claims.

Second, for the sake of argument, even if one were to combine these references, the disparate objectives of Abrashkevich and Kekre would not lead one of ordinary skill in the art at the time the invention was made to what is recited by claim 1. As recited by claim 1 as amended, a timestamp indicates when a memory block is requested or freed; in other words, claim 1 recites storing in a memory block the time when (1) the memory block was requested, and thus first potentially made available for use; and (2) when the memory block was freed, and thus no longer available for use. On the other hand, Kekre addresses the ability to move between versions of data saved at any time the memory already was available – between the times when the memory was requested or freed. In other words, Kekre and claim 1 are directed to separate, non-overlapping periods of time. Kekre is directed to entirely different objectives than what is recited by claim 1, and thus cannot teach what is recited by claim 1.

Third, Kekre never really contemplates allocation or freeing of memory blocks, Kekre fails to teach literal limitations recited by claim 1. The Manual of Patent Examining Procedure requires that “the prior art reference (or references when combined) must teach or suggest all the claim limitations.” *See* MPEP § 706.02(j). Kekre includes some discussion of “allocating,” with regard to allocating “a new region . . . on a cache volume” (Kekre, Column 11, Line 63) or allocating “a new entry in the B+- tree” (Kekre, Column 11, Line 28). However, neither of these passages, or any other passages of Kekre, address the allocation of a memory block, let alone the

freeing of a memory block. Accordingly, adding Kekre to Abrashkevich still leaves limitations of claim 1 that are not described by the references.

Fourth, although Kekre does contemplate the release of data (although not the release of memory), it does not describe timestamping or otherwise tracking a time once the data is released. Kekre describes that, after a certain amount of time has passed or if a user so chooses, that its data history may be released. “If the user wants to delete the history . . . delete all region copies with timestamp $>T$, 25, (this basically means returning the data blocks to free pool).” Kekre, Column 19, Lines 23-27. From this, one might infer that a timestamp of when Kekre’s data was last written or read will have a bearing on when it is freed – any timestamp indicating reading or writing before a designated point in time “T” is released.

However, *there is absolutely no mention of then applying a timestamp to that data to indicate when that data was freed*. The data freed may have a timestamp a millisecond older than “T,” or years older than “T” – but Kekre does not describe applying a timestamp to that data. On the other hand, claim 1 expressly recites that a timestamp specifically directed to recording the time when a memory block is allocated or freed, by

storing a timestamp within the allocable memory block, wherein the timestamp is configured to indicate ~~indicates~~ a time when ~~one of the requesting and the freeing of the allocable memory block is performed~~:

the requesting of the allocable memory block is performed unless the timestamp indicates a time when the allocable memory block is freed; and
the freeing of the allocable memory block is performed unless the timestamp indicates a time when the allocable memory block is requested,

(Claim 1; markings to show additions and deletions removed). If anything, in failing to describe applying a timestamp to data once the data is no longer wanted, Kekre teaches away from what is recited by claim 1. Again, Kekre cannot make up for the shortcomings of Abrashkevich.

In sum, Kekre fails to make up for the admitted shortcomings of Abrashkevich in several ways. First, no reasonable person would have combined Kekre's versioning system with Abrashkevich's memory diagnostic system. Second, Kekre focuses on such a different object than either Abrashkevich or claim 1 that merely knowing of both references would not lead one to combine them to reach what is recited by claim 1. Third, because Kekre fails to associate timestamps with times at which a memory block is requested or freed, the combined references fail to teach all limitations of claim 1. Fourth, Kekre's failure to describe applying a timestamp after memory is freed teaches – if anything – away from what is recited by claim 1. Thus, for at least these reasons, the combined references do not support an obviousness rejection against claim 1. The rejection against claim 1 should be withdrawn, and claim 1 should be found to be in condition for allowance.

Without belaboring the point, independent claims 11 and 21 include the same limitations as claim 1 with regard to the storing of timestamps. All the foregoing arguments with respect to claim 1 apply equally to independent claims 11 and 21. Accordingly, the obviousness rejection must be withdrawn against claims 11 and 21, and both claims should be found in condition for allowance.

Claims 2-9, 12-19, and 22-29 depend from and apply additional limitations to the respective independent claims from which each depends. Thus, claims 2-9, 12-19, and 22-29 are patentable for at least the same reasons for which claims 1, 11, and 21 are allowable. Thus, in

sum, all pending claims are in condition for allowance, and a notice of allowance is respectfully requested.

CONCLUSION

In view of the foregoing amendments and remarks, all pending claims are believed to be allowable and the application is in condition for allowance. Therefore, a Notice of Allowance is respectfully requested. Should the Examiner have any further issues regarding this application, the Examiner is requested to contact the undersigned attorney for the applicants at the telephone number provided below.

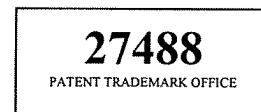
Respectfully submitted,

MERCHANT & GOULD P.C.



Frank J. Bozzo
Registration No. 36,756
Direct Dial: 206.342.6294

MERCHANT & GOULD P.C.
P. O. Box 2903
Minneapolis, Minnesota 55402-0903
206.342.6200



APPENDIX:
REPLACEMENT DRAWING SHEET